# Generating on-board diagnostics of dynamic automotive systems based on qualitative models*

Fulvio Cascio[1], Luca Console[2], Marcella Guagliumi[3], Massimo Osella[1], Andrea Panati[2], Sara Sottano[1], Daniele Theseider Dupré[2]

(1) Centro Ricerche Fiat, Strada Torino 50,
10043 Orbassano (Torino), Italy, email: {f.cascio,m.osella,s.sottano}@crf.it
(2) Dip. Informatica, Università di Torino,
Corso Svizzera 185, 10149 Torino, Italy, email: {lconsole, panati, dtd}@di.unito.it
(3) Magneti Marelli Electronic Systems Division,
Viale Carlo Emanuele II 118, 10078 Venaria Reale (Torino), Italy,
e-mail: Marcella.Guagliumi@venaria.marelli.it

**Abstract**

On-board diagnostic systems play an important role in the current generation of cars and will play an increasingly important role in the next future. The design of on-board diagnostic systems is a challenging problem under several points of view. In this paper we discuss the experience we made on such a problem within the VMBD project. In particular, we discuss an approach which tries to reconcile two goals: satisfying all the requirements and constraints imposed by the on-board application, and exploiting the advantages of the model-based approach as much as possible. The approach is based on qualitative deviation models for the automatic derivation of on-board diagnostics based on decision trees. In the paper we use a specific application, the Common Rail fuel delivery system, as a concrete example, briefly discussing the on-board diagnostics we designed for such a system and its prototype implementation and demonstration.

## Introduction

The aim of this paper is to discuss the experience we made, within the VMBD project, on the design of on-board diagnostic systems for automotive applications. VMBD aims at demonstrating the utility of model-based diagnosis in the automotive domain for both on-board and off-board diagnosis. This goal has been achieved through the definition of a general formalism for building a library of (qualitative) models of components, the definition of an architecture for off-board and on-board diagnostic problem solving, and the experimentation on three guiding applications. Within the VMBD project, our group mainly focused on on-board diagnostic problem solving, which is interesting under several points of view. First of all, there is a fixed set of sensors, and tests can hardly be performed. Further, the on-board context imposes specific hardware requirements (memory and computing power) on the design of the diagnostic system. However, the response time should be short, especially for safety critical systems, and should concentrate on taking the appropriate action, without necessarily identifying the fault. Finally, systems to be diagnosed on-board are in almost all cases dynamic feedback systems with an active control which is in most cases performed by the ECU (Electronic Control Unit) software, which can often compensate for faults.

Thus, these systems are a challenging application for testing state-of-the-art model-based diagnostic techniques. In the paper we analyse how the requirements sketched above can be taken into consideration in the design of a diagnostic architecture for on-board systems with the following goals:

1. producing a diagnostic system that can be conceivably used on-board, given the technologies that will be feasible on cars in the next few years;
2. exploiting the advantages of the model-based approach as much as possible.

As a running example we use one of the guiding applications in the VMBD project: the Common Rail fuel delivery system. It has all the features discussed above (safety critical, real time recovery, dynamic feedback with active control) and is interesting for both on-board

* This work was partially supported by the European Commission, DG XII (project BE 95/2128, "VMBD"). VMBD (Vehicle Model-Based Diagnosis) is a Brite-Euram project involving the following partners: Daimler Benz, Centro Ricerche Fiat, Volvo, Bosch, Magneti Marelli, Genrad, Dassault Electronique, Università di Torino, Université Paris XIII (Paris Nord), University of Wales at Aberystwyth.

diagnosis (performing recovery actions in the presence of malfunctions) and off-board diagnosis. Moreover, the Common Rail involves hydraulic, electric and electronic components, which are, at least in part, similar to those used in other systems. Thus, generic models of these components can be reused in different systems (as regards VMBD, the Common Rail has a number of common components with another application, the Distributor Type Injection). In the paper:

1. We first introduce the Common Rail system used as a guiding application, and the models based on qualitative deviations we adopted for it, motivating the suitability of this kind of models for our goals.
2. We then discuss the diagnostic approach we adopted. In particular, we briefly recall a simulation-based approach to deal with dynamic behavior; we then analyse the problem of on-board diagnosis, motivating and discussing an approach which uses the model-based system for automatically synthesizing efficient on-board diagnostics in the form of decision trees.
3. We finally provide an example of model-based diagnosis results used in the generation of on-board diagnostics for the Common Rail and we sketch the prototype demonstration on board of a Lancia car.

## The Common Rail system

The Common Rail fuel injection system (Stumpp & Ricco 1996) for direct injection diesel engines is designed in order to be able to control the injection pressure, as well as injection timing, which allows better engine performance and lower noise and emissions. To this end, pressurised fuel is stored in the rail and its pressure is controlled by the Electronic Control Unit (ECU) through a pressure regulator.

In more detail, the purpose of the main components (see figure 1) is as follows. The high pressure pump delivers fuel to the rail, which, together with the high pressure pipes between the high pressure pump and the injectors, behaves as an accumulator. The pressure regulator, an overflow valve controlled by the ECU, varies pressure in the rail. When the driving current is increased, the regulator closes an orifice. This determines a decrease of the overflowing fuel amount and a consequent increase of the pressure in the rail. The overflowing fuel is returned to the tank.

The ECU controls the fuel injection system. The target pressure value for the fuel pressure is determined given the engine operating conditions. If the rail pressure, measured by the pressure sensor, deviates from the target value, the command to the pressure regulator is varied in order to reduce the difference between the measured pressure and the target value. Injectors also receive commands from the Electronic Control Unit, which computes both the amount of fuel to be injected and injection timing. In case of faults, possible actions to be taken by the ECU are:

- limiting performances, i.e. lowering the maximum value for the rail pressure (which also limits the achievable acceleration);
- switching to a "limp home" mode where the system variables are forced to be in idle mode. In this way the rail pressure is low, and the engine speed is kept constant, while still allowing the driver to reach e.g. a service bay;
- stopping the engine when some dangerous fault is suspected.

The fuel delivery subsystem has been modified in the test car in the following ways:

- A sensor in the low pressure subsystem (between the filter and the high pressure pump) has been introduced; it detects whether the pressure is sufficient to deliver fuel to the high pressure pump.
- The effect of introducing a "virtual" (i.e., software) sensor based on engine speed has been considered. Such a type of sensor has been developed by Centro Ricerche FIAT for torque measurement (to detect anomalous injection amount and timing), i.e. for computing whether a cylinder provides a significantly different torque with respect to other cylinders. This is a "virtual" sensor since it provides an indirect measure based on actual measures. In principle, there would be no need for such a concept of virtual sensor in a model-based diagnostic system; this one, in particular, could be substituted with just the engine speed sensor and models of combustion and of the engine dynamics. But since such models would be much more complex than the rest, the virtual sensor abstraction has been preferred as a more viable solution. Admittedly, this is an ad-hoc abstraction; general and principled ways of abstracting models have not been the purpose of this work.
- Hardware has been included for introducing in the system some of the faults considered in the models; in particular, for switching off the electric pump in the low pressure subsystem, for switching off the PWM (pulse width modulation) command to the pressure regulator (so that it remains open), or for supplying or not supplying one injector with current, so that it remains always open, or always closed, regardless of the opening commands from the ECU.

## Qualitative deviation models

Applying model-based diagnosis to dynamic controlled systems is one of the main focuses of research in the field (e.g. (Chantler *et al.* 1996; Loiez & Taillibert 1996; Mosterman & Biswas 1996; 1997; Malik & Struss 1996; Struss 1997)). In (Malik & Struss 1996), in particular, an approach based on qualitative deviations has been used. The system is modeled in terms of differential equations that include appropriate parameters for components, whose values correspond to different (correct or faulty) behavior modes of the component. From these equations, corresponding equations for qualitative deviations are derived:

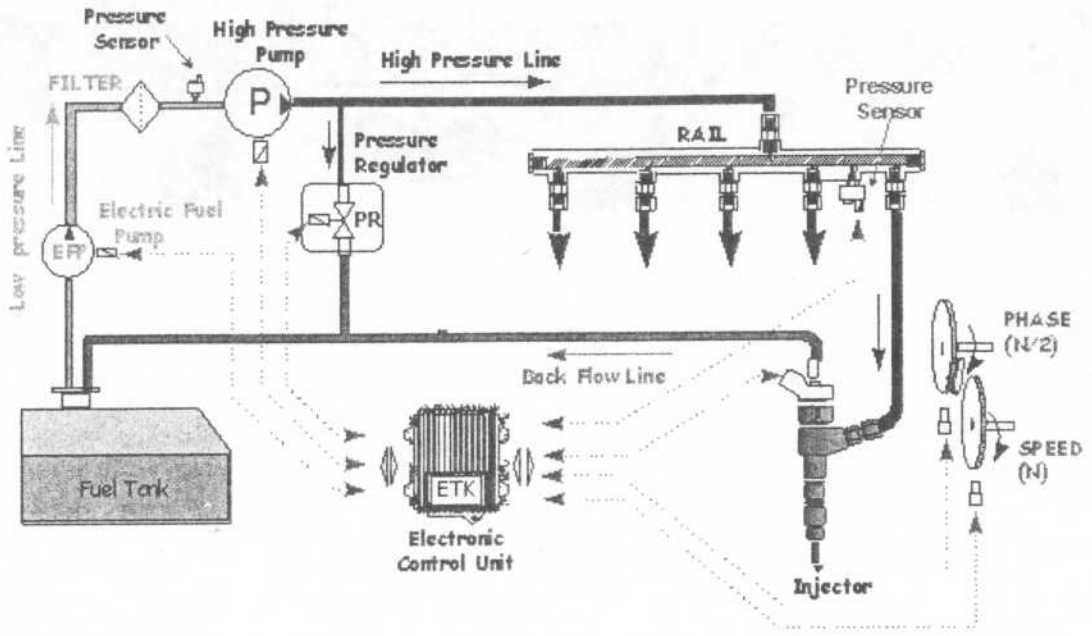


Figure 1: The Common Rail fuel injection system for the 5-cylinder car used for demonstration, with an additional pressure sensor in the low pressure part. One of the 5 injectors is enlarged for better readability.

- for each variable x, its deviation $\Delta x(t)$ is defined as $\Delta x(t) = x(t) - x_{ref}(t)$, where $x_{ref}(t)$ is a reference behavior;
- from any equation $A = B$, the corresponding equation $\Delta A = \Delta B$ is derived;
- finally, the corresponding qualitative equation $[\Delta A] = [\Delta B]$ is derived; it equates the signs of the two deviations. There are rules for expressing this equation in terms of signs of deviations of individual variables rather than expressions.

Similar ways for deriving qualitative models from quantitative equations have been proposed in (Gallanti *et al.* 1989; Biswas, Kapadia, & Yu 1997). This form of qualitative modeling has been chosen for this system, especially because the fuel pressure in the rail is rapidly varying, according to the position of the accelerator pedal and a number of other inputs, therefore a *normal* range of values cannot be given. This means that reasoning in terms of absolute values would be very difficult. A main problem in applying this approach is however the choice of the reference behavior. The choice in (Malik & Struss 1996), which is shown to be useful at least in some cases, is to have a steady state as the reference behavior. We regarded the reference behavior as the evolution of the system when all components are not faulty. As we will discuss later, this choice has strong influences on the way diagnosis is activated, i.e. how it is detected that there is a fault in the system. The model used for our experiments is shown in the block diagram in figure 2; some simplifications have been done with respect to a model that would be derived from models of individual components:

- the low pressure part of the system has been abstracted to a single component; similarly for the rail and the high pressure pipes;
- the "torque measurement" virtual sensor has been attached directly as an output to the injector component, rather than introducing the engine component.

Variable names on the arcs correspond to pairs of interface variables of components, which are imposed to be equal by the connection. The meaning of such variables is the following:

- $f\_lowp$: flow from the low pressure subsystem (into the high pressure pump)
- $p\_lowp$: pressure in the low pressure subsystem
- $lowp\_obs$: sensor reading for the low pressure subsystem
- $f\_pump$: outflow of the high pressure pump
- $p\_rail$: pressure in the rail
- $p\_obs$: reading of the pressure sensor in the rail
- $PWM$: actuation command to the pressure regulator
- $f\_pr$: flow through the pressure regulator
- $f\_inj$: flow through the injectors
- $tm$: torque measurement (virtual sensor)

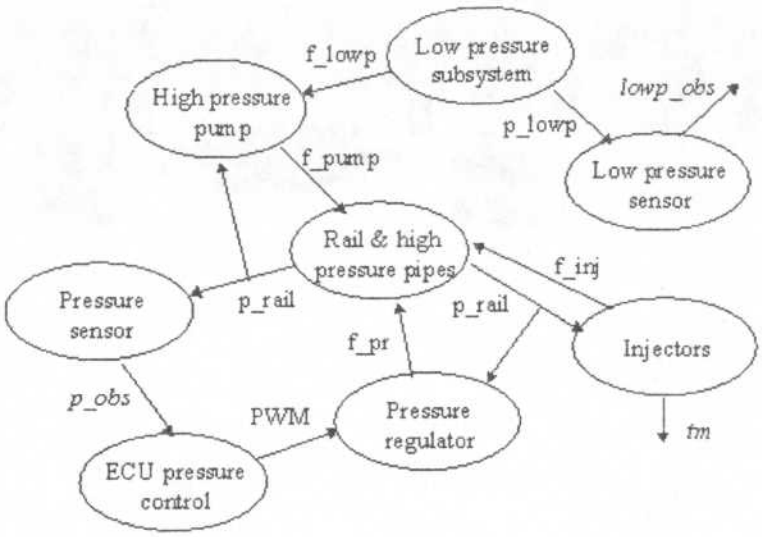


Figure 2: Diagram of the model of the Common Rail.

As an example of qualitative deviation modeling, the following qualitative deviation equation is included in the model of the high pressure pipes and rail: $\partial\Delta p\_rail = [\Delta f\_pump]$ - $[\Delta f\_pr]$ - $[\Delta f\_inj]$ - $[\Delta f\_leak]$ where:

- $f\_leak$ is the flow through possible leaks; it is a parameter of the component and its reference value is 0; the "ok" mode for the component imposes $[\Delta f\_leak] = 0$, that is, there are no leaks, while the only fault mode considered for the component, "leaking", imposes $[\Delta f\_leak] = +$ ;
- $\partial\Delta p\_rail$ is the derivative of the deviation (or, equivalently, the deviation of the derivative) of the rail pressure. The equation implies that if the balance of flows becomes different (e.g. smaller) from the expected one (e.g., due to a smaller- than-expected inflow from the pump, or a higher-than-expected outflow through the regulator, or the injectors, or a leak), then the derivative of the rail pressure deviates (e.g. becomes smaller). This will make the rail pressure itself deviate (e.g. become smaller) with respect to the expected trend.

The list of fault modes considered for the components comes from the system FMEA and is the following.

The high pressure pump (*hp_pump*) has fault modes *blocked*, with obvious meaning, and *insuff* which means it has a reduced efficiency. The high pressure pipes and rail (*hp_pipes_rail*) have *leak* as the only fault mode. The pressure regulator (*p_regulator*) has the fault modes *blocked closed* and *blocked open*. The *injectors* have fault modes *blocked open* and *blocked closed*, as well as *recycle low* and *recycle high* which correspond to an abnormal amount of fuel returned to the tank (part of the fuel flowing through the injectors is in fact going back to the tank). The low pressure subsystem (*lp_system*) has the only fault mode *insuff* which means it does not deliver enough fuel; this includes a fault of the electric fuel pump. For the sensor in the low pressure system no fault modes are considered, for the sensor in the high pressure part, the faults *low*, *high* and *blocked* are considered. Usually, sensor faults can be identified by an implausible sensor reading, i.e. through a range check. This could be modeled introducing additional qualitative values for deviations, but in the running example it has not been done for the sake of simplicity (as a consequence, in the diagnostic results discriminating sensor faults from some other faults will not be possible).

## Diagnostic strategies

As we noticed before, we are interested in both off-board and on-board diagnosis. While off-board diagnosis can take full advantage of the model-based approach, on-board diagnosis requires some further considerations and taking into account some important practical constraints. First of all, in the on-board case the diagnostic system must react promptly to anomalies and must run fast in order to interpret the anomalies and, which is most important, in order to take an appropriate recovery action. This means also that the on-board diagnostics should be focused only on performing recovery actions, rather than on the actual isolation of the fault(s). However, keeping track of more detailed information about the fault(s) that occurred can be extremely important, as it is a valuable information to be passed to the diagnostic system that will be used in the workshop to actually locate and identify the fault. Moreover, only a few measurements can be available on-board and the possibility of performing tests/probes is

very limited. Finally, even though integrating the diagnostic system within the ECU is not one of the goals of VMBD, we must build a diagnostic system that takes into account the kind of hardware that is available on board (or that will be presumably available in the next years). For example, the total amount of memory available on the ECU (for both control and diagnosis) of the test car is 64K. Therefore, we must reconcile two goals that are, at least partially, conflicting:

1. the aim of showing that the model-based technology provides interesting advantages and can be exploited and have an impact also for on-board applications;
2. the constraint of not requiring a major revolution in the hardware (and software) technologies currently used on-board.

Such requirements show that the direct adoption of model-based diagnosis on board can be problematic. Moreover, using the full power of the model-based approach on board could be unnecessary, especially as regards selecting measurements and tests, since little space for performing additional measurements and tests is available.

The on-board diagnostic task can be better performed using simpler technologies, such as "pattern-action" rules or decision trees. Such technologies in fact can be easily implemented on current ECUs and could thus be practically used immediately. However, we believe that the model-based approach can nevertheless play a fundamental role in automatically synthesizing the simpler knowledge base and diagnostic strategy to be implemented on-board.

In the two following sections we discuss the definition of diagnosis for dynamic systems we adopted and how such a definition can be exploited for compiling automatically the on-board diagnostic system.

## Diagnosis of dynamic systems

In (Theseider Dupré & Panati 1998) several alternatives are considered for defining diagnosis of dynamic systems, including state-based diagnosis (Malik & Struss 1996; Struss 1997) and some simulation-based definitions which take into account the discontinuity in the behavior associated with abrupt faults, i.e. the sudden transition of a component from the correct mode of behavior to a faulty behavior. It is shown that, given additional knowledge on causality in the system, reasoning on such transitions and using simulation can lead to reducing the set of state-based diagnoses, under increasingly restrictive assumptions on the observability of the system, i.e. on fault detection.

The assumptions on fault detection are important. Differently from (Malik & Struss 1996), we assumed as a reference behavior the correct behavior of the system; that is, a deviation should be detected (and then diagnosis activated) when the system deviates from its expected behavior. But, given noise in measurements and imprecise knowledge about the quantitative behavior of the system under correct behavior, it cannot be assumed that arbitrarily small deviations can be detected. Moreover, some assumptions must be done, based on observation on the real system, on the relative speed on which deviations of different variables can be detected.

As we shall briefly describe in the final section, in the Common Rail application different approaches have been applied for detecting deviations for the different variables.

The simulation-based approach in (Theseider Dupré & Panati 1998) has been used as a definition of diagnosis, and it has been assumed that all deviations can be detected "at the same time", in the sense that no qualitative state is missed by fault detection (this is similar to the "gapless observations" in (Malik & Struss 1996)).

Therefore, diagnosis is activated for the first qualitative state where non-zero deviations are detected. Given the set of observations OBS for such a state, the consistency of a mode assignment $F$ with OBS (i.e. its being a state-based diagnosis for the observations) is not sufficient to consider F a diagnosis. In fact, it is also imposed that this state ($S_n$ in figure 3) must be reachable from an initial state ($S_0$) where all components are "ok" and all deviations are zero, through a sequence of states satisfying the following conditions:

- The first state ($S_1$) derives from causal knowledge about the system that gives the result of "injecting" the fault F into the initial state. For example, consider the sample equation given above for the rail component: $\partial\Delta p\_rail = [\Delta f\_pump] - [\Delta f\_pr] - [\Delta f\_inj] - [\Delta f\_leak]$ If the rail starts leaking, i.e. the value of $[\Delta f\_leak]$ changes from 0 (in $S_0$) to positive (in $S_1$), some other change must occur in order for the equation to hold in $S_1$; but due to the causality in the system, the only change that can occur in the first state is $\partial\Delta p\_rail$ becoming negative, which will make $[\Delta p\_rail]$ negative (i.e. the rail pressure smaller than expected) in $S_2$.

  Changes to the other variables will occur later due to feedback in the system: in fact, flows depend on pressure, and therefore they will actually change (or, better, deviate from their expected value), but only after the pressure itself deviates. Note that flows depend on pressure both "naturally" and due to the pressure control system: the ECU will command the pressure regulator to close more than it should (and then reduce the outflow) when the pressure (read through the pressure sensor) deviates negatively.
- The subsequent states, up to the final state ($S_n$) consistent with the observations, are each one a successor state of the previous one according to qualitative simulation.
- The sequence $S_1, \ldots, S_{n-1}$ cannot contain qualitative states with deviations that would have been detected before the one (or ones) that actually activated diagnosis. What this actually means depends on the assumptions on fault detection. In principle, it could mean that no deviation of observable variables should

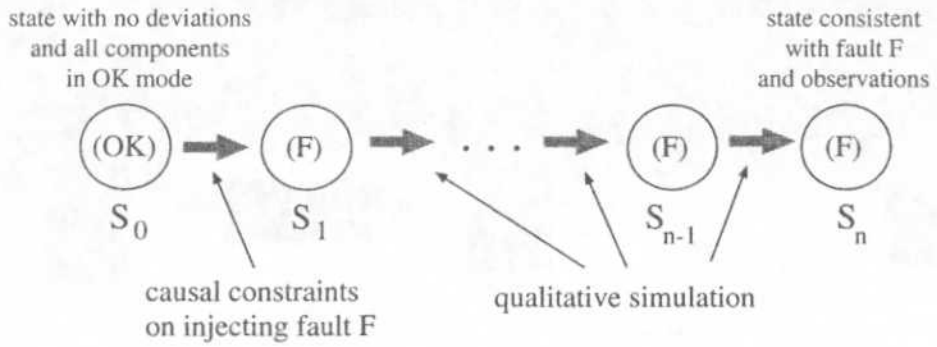


Figure 3: Simulation-based diagnosis.

occur in $S_1, \ldots, S_{n-1}$. In practice, this strong condition should be relaxed, at least to allow in the sequence qualitative states with observable deviations if they are states that only hold for a time point and not for an interval. An example of such a state is the one mentioned above where $\partial\Delta p\_rail$ is negative but $[\Delta p\_rail]$ is zero. Moreover, since detecting deviations requires elaboration of actual data, the time necessary for detecting deviations of different variables may be different.

For a formal description the reader is referred to (Theseider Dupré & Panati 1998).

## Compiling on-board diagnostics

We already discussed the problems that have to be faced in the design of on-board diagnostic systems and we motivated the choice of precompiling knowledge for the on-board system.

In this section we discuss in more detail how the compilation can be performed taking advantage of the model-based diagnostic approach discussed in the previous section. Examples will be provided in the next section. As we noticed before, the basic type of knowledge needed in the on-board system is a set of associations between patterns of values of observable parameters and recovery actions. We decided to implement such associations and the classification process to be used for selecting the best action in a given situation as decision trees. In fact, a decision tree interpreter can be implemented very efficiently and easily on any hardware support, using little memory: therefore, these methodologies could be easily used on current ECUs. Moreover, there are well known and established algorithms for building trees from examples (see e.g. algorithms such as ID3 (Quinlan 1986)). In our case an example is an association between a pattern of values of the observable parameters and the corresponding action (and diagnoses). What is most interesting is that the set of examples can be produced automatically using a diagnostic engine that is directly based on models, and could be the same one used for off-board diagnosis. We thus have the following scheme for the generation of the on-board diagnostic system:

1. Determine a set of significant cases that must be faced on-board. In case the number of parameters and the set of qualitative values of the parameters are small, one may even consider an exhaustive set of cases.
2. Run the off-board model-based diagnostic system on each one of the cases, computing the set of candidate diagnoses for each case and the corresponding recovery action.
3. Use a learning algorithm to derive the decision trees from the examples produced in the previous items.

Let us analyse in more detail the last step above. The input to the learning algorithm is a table with one row for each one of the significant cases selected in step (1). The columns correspond to the observable parameters that will constitute the nodes of the decision tree; two special columns store the decision (recovery action) and the set of candidate diagnoses for each one of the cases, computed in step (2). The decision tree is built by selecting, at each step, the observable whose values best discriminate between the possible decisions (a measure of the discrimination power is computed using entropy). At the initial step the selection is made on the whole table and this leads to selecting the observable $A$ which is the root of the tree. Such a node has one descendant for each possible value of $A$. The descendant corresponding to the value $a_i$ of $A$ is a decision in case all the examples in the table for which $A = a_i$ correspond to the same decision. Otherwise, a subtree is built using as examples those for which $A = a_i$.

The decision tree representation has some advantages with respect to other forms of associational knowledge. In case all the values for the observations are known to the ECU, the decision tree is anyway more compact than a lookup table. Pattern-matching rules could also be used. However, the decision tree is better generalized to the case (which is conceivable, even if not widely used in ECUs in the automotive domain) where some decision requires an active test to be performed. In fact, the decision tree can provide an order on data to

| Observations | | | Diagnoses | Actions |
|---|---|---|---|---|
| $\Delta lowp\_obs$ | $\Delta p\_obs$ | $\Delta tm$ | | |
| - | - | 0 | *Lp_system insuff* | **Performance limitation** |
| 0 | - | 0 | *Pressure_regulator blocked_open* | *Performance limitation* |
| | | | *Hp_pipes_and_rail leak* | **Limp home** |
| | | | *Hp_pump blocked* | *Performance limitation* |
| | | | *Hp_pump insuff* | *Performance limitation* |
| | | | *Sensor low* | *Open loop* |
| | | | *Sensor blocked* | *Open loop* |
| | | | *Injectors recycle_high* | *Performance limitation* |
| 0 | - | + | *Injectors blocked_open* | **Stop** |
| 0 | 0 | - | *Injectors blocked_closed* | **Go** |
| 0 | + | 0 | *Pressure_regulator blocked_closed* | **Stop** |
| | | | *Sensor low* | *Open loop* |
| | | | *Sensor blocked* | *Open loop* |
| | | | *Injectors recycle_low* | *Go* |

Table 1: Combinations of observations with corresponding single fault diagnoses and recovery actions. The selected action is shown in boldface font.

be used and, more importantly, the tests are performed only when actually necessary, i.e. when values of other data are not sufficient for discriminating between the recovery actions. Cost of tests could be used, together with entropy, to select the best observation at each step in the tree generation.

## An example

In this section we provide an example of the process for compiling decision trees according to the strategy defined in the previous section. We consider the three following observables in the common rail model: the rail pressure, the pressure in the low pressure subsystem and the torque measurement. In particular, we consider the qualitative values for the deviations of these measurements, i.e., respectively, $[\Delta p\_obs]$, $[\Delta lowp\_obs]$ and $[\Delta tm]$.

We then consider (see table 1) the combination of qualitative values for these variables with the corresponding single fault diagnoses computed with the dynamic diagnosis approach described before, and the recovery action corresponding to each one of the cases. Cases not listed have no single fault diagnosis. When there is more than one candidate diagnosis associated with a set of observations (which means that the candidates cannot be discriminated with the available observables), the recovery action associated with the set of candidates (the one in boldface) is the most critical one, that is the one associated with the most critical fault. The cases considered include the 4 faults that can be artificially introduced in the demonstrator car, which are: *lp_system insuff*; *pressure_regulator blocked open*; *injectors blocked open*; *injectors blocked closed*. Notice that in three cases the actual fault can be identified as the only single fault diagnosis; in other cases, there are multiple diagnoses but all of them are sensible diagnoses for the given observations.

It is important to notice that the use of the dynamic diagnosis approach outlined in a previous section is essential to the results in table 1.

In fact, applying state-based diagnosis (in the same context, i.e. with the same interpretation of deviations) provides some unexpected solutions: e.g. for the second case in table 1, i.e. $\Delta lowp\_obs = 0$, $\Delta p\_obs = [-]$, $\Delta tm = 0$: *Pressure_regulator blocked_closed*, *Sensor high*, *Injectors recycle_low*, and the empty diagnosis (all the components ok).

The presence of spurious diagnoses could lead to selecting an unnecessary restrictive action: in the example, **Stop** would be selected instead of **Limp home**, because of the spurious diagnosis *Pressure_regulator blocked_closed*.

Moreover, the fact that the "all ok" mode is consistent would contradict fault detection: some "abnormal" observation is detected, but the system can anyway and it would be assumed to be ok by any sensible preference criterion for diagnoses; e.g. we have used cardinality, preferring single fault diagnoses, and this would make the "all ok" candidate the preferred diagnosis.

Given this table, the process for compiling the decision tree can be started. The result is the decision tree shown in figure 4, which indeed captures the intended behavior for the on-board diagnostic system.

It is worth noting that in this case the diaghostic system is better than the diagnostic procedures currently used in the Common Rail system, which, however, cannot use the measurements of the pressure in the low pressure system and the torque (but uses other pieces of information and plausibility checks concerning the values of internal variables in the ECU). This is an interesting result of the adoption of our approach since the experimentation with the model allowed us to study the effect of adding additional sensors. The example shows a compromise in which better on-board discrimination on the recovery action to be performed can be obtained with additional sensors.

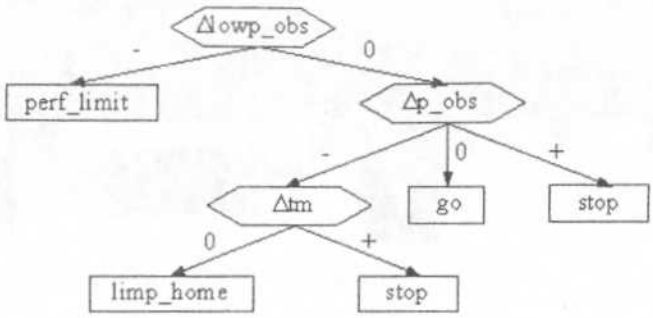


Figure 4: Compiled decision tree.

## Prototype implementation and demonstration

In this section we briefly discuss the implementation of the diagnostic system described in the previous sections. In order to have a first prototype at an early stage and make experiments as regards both modeling and the diagnostic strategy, we implemented a first version of the diagnostic system using ECLiPSe, a constraint logic programming language. These kinds of languages are in fact well suited for rapid prototyping, and constraint propagation for simulating models can be implemented very easily. One strategy has also been implemented in C++. As regards the generation of the on-board decision trees we used an implementation of the standard ID3 algorithm (Quinlan 1986).

For the demonstration, a prototype Lancia car with a pre-series 2.4 Litres Common Rail engine has been equipped with appropriate hardware and software for data acquisition (see figure 5). ETK is a hardware interface attached to the ECU, providing access to the controller data bus; MAC is a protocol conversion box; INCA- PC (by ETAS) is the software for acquisition and display of data, running on a portable PC together with the on-board diagnostic system developed for VMBD.

We regard this system as "on-board" even if it does not run on the ECU itself; this solution was out of the scope of the project, but it is already feasible, even given the current ECU hardware limitations. Such a system includes:

- A module for the conversion of signals (acquired by INCA) into qualitative deviations; this is easily done for the sensor in the low pressure part of the system, where the pressure has a nominal range of values, while for the pressure sensor in the high pressure part a comparison is made on the desired pressure computed by the ECU, relying on an approximation on the expected behavior of the actual pressure.
- The decision tree interpreter.

Experiments on the demonstrator system were successful in checking that, when one of the faults was introduced in the system, the appropriate values for qualitative deviations were acquired. Therefore, since table 1 contains sensible diagnoses for each case, the system can suggest the most appropriate action.

## Conclusions

In this paper we discussed our experience in the design of on-board diagnostic systems for automotive domains. We analysed the peculiarities of on-board applications, discussing their requirements and the consequences that the requirements have on the design of the diagnostic system. In doing that we took into account a further main goal: trying to exploit as much as possible the advantages of the model-based approach also for on-board applications. The results discussed in the paper is a compilation-based approach: the model-based approach is used for simulating diagnostic situations; the results of simulations (diagnoses and actions corresponding to a set of observations) are used by a learning algorithm to derive the decision trees that will form the on-board diagnostic system.

We claim that this approach has significant advantages on other ones. With respect to the adoption of on-board model-based systems, our approach has the advantage of being close to be implemented with current ECU technology, even for a dynamic, controlled system. With respect to a manual generation of the decision tree (as it is currently done in many diagnostic systems), on the other hand, we have all the advantages coming from relying on reusable models, which can significantly reduce the effort in generating the tree for a new system. Moreover, the experimentation with the model-based approach can lead to studying how to improve the diagnosability of the system, e.g. by adding extra sensors (as it is indeed shown by our example).

The idea of compiling diagnostic rules from examples has been used in other approaches, even if with some differences. In (Price *et al.* 1996), which shares several characteristics of our approach, models are used for generating FMEA, which is in turn used for generating diagnostic trees for off-board diagnosis. Previous approaches used qualitative models for running a set of simulations and induce diagnostic rules from

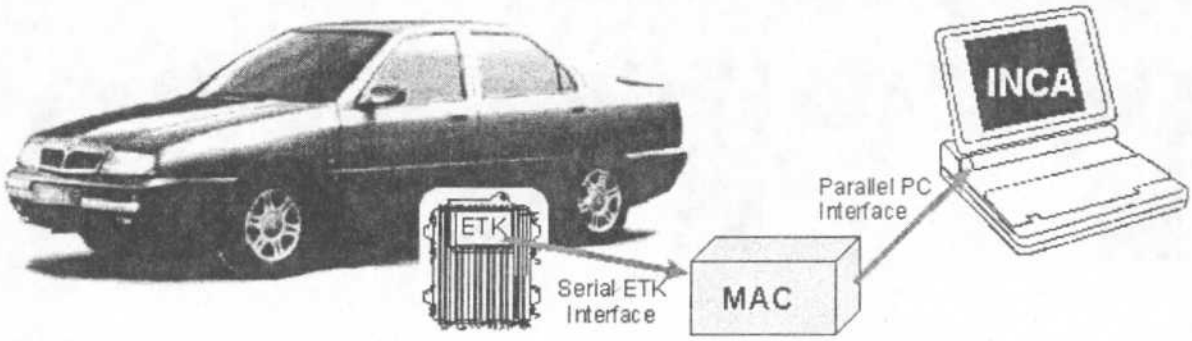


Figure 5: Equipment for the demonstrator.

the results of the simulations (see, e.g., (Mozetic 1991; Pearce 1988)). Thus there are two main differences with respect to the approach presented in this paper: they run simulations of combinations of faults, while our examples are sets of observables corresponding to the diagnostic problems to be solved (which is more focused); they perform induction from examples while we synthesize decision trees. A different approach based on explanation based learning has been adopted in (Steels & Van de Velde 1985): they start from a single example and try to induce rules from the solution to the example. This, however, is a critical step in the induction of diagnostic rules, as discussed in detail in (Console, Portinale, & Theseider Dupré 1996).

## Acknowledgements

We wish to thank all partners in the VMBD project, and especially Bosch for cooperation on the Common Rail demonstrator. We also thank Peter Bidian, Peter Struss, Reinhard Weber and the anonymous reviewers for useful comments on earlier versions of this paper.

## References

Biswas, G.; Kapadia, R.; and Yu, X. 1997. Combined qualitative-quantitative steady-state diagnosis of continuous-valued systems. *IEEE Trans. on Systems, Man and Cybernetics* 27(2):167–185.

Chantler, M.; Daus, S.; Vikatos, T.; and Coghill, G. 1996. The use of quantitative dynamic models and dependency recording for diagnosis. In *Proc. 7th Int. Work. on Principles of Diagnosis*, 59–68.

Console, L.; Portinale, L.; and Theseider Dupré, D. 1996. Using compiled knowledge to guide and focus abductive diagnosis. *IEEE Transactions on Knowledge and Data Engineering* 8(5):690–706.

Gallanti, M.; Roncato, M.; Stefanini, A.; and Tornielli, G. 1989. A diagnostic algorithm based on models at different levels of abstraction. In *Proc. 11th IJCAI*, 1350–1355.

Loiez, E., and Taillibert, P. 1996. Polynomial temporal band sequences for analog diagnosis. In *Proc. 7th Int. Work. on Principles of Diagnosis*, 139–146.

Malik, A., and Struss, P. 1996. Diagnosis of dynamic systems does not necessarily require simulation. In *Proc. 7th Int. Work. on Principles of Diagnosis*, 147–156.

Mosterman, P., and Biswas, G. 1996. An integrated architecture for model-based diagnosis of dynamical physical systems. In *Proc. 7th Int. Work. on Principles of Diagnosis*, 167–174.

Mosterman, P., and Biswas, G. 1997. Monitoring, prediction and fault isolation in dynamic physical systems. In *Proc. AAAI 97*, 100–105.

Mozetic, I. 1991. Hierarchical model-based diagnosis. *Int. J. of Man-Machine Studies* 35(3):329–362.

Pearce, D. 1988. The induction of fault diagnosis systems from qualitative models. In *Proc AAAI 88*, 353–357.

Price, C.; Wilson, M.; Timmis, J.; and Cain, C. 1996. Generating fault trees from FMEA. In *Proc. 7th Int. Work. on Principles of Diagnosis*, 183–190.

Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning* 1:81–106.

Steels, L., and Van de Velde, W. 1985. Learning in second generation expert systems. In Kowalik, R., ed., *Knowledge-based Problem Solving*. Prentice Hall.

Struss, P. 1997. Fundamentals of model-based diagnosis of dynamic systems. In *Proc. IJCAI 97*, 480–485.

Stumpp, G., and Ricco, M. 1996. Common rail - an attractive fuel injection system for passenger cars. *J. Society of Automitive Engineers*.

Theseider Dupré, D., and Panati, A. 1998. State-based vs simulation-based diagnosis of dynamic systems. In *Proc. DX 98, 9th Int. Workshop on Principles of Diagnosis*.